

Matlab Basics

Yaara Erez

MRC Cognition and Brain Sciences Unit

November 2012

MatLab – Matrix Laboratory

- Programming environment based on matrix representations.
- Mainly useful for data analysis, simulations (research, engineering).
- Contains a large set of ready-to-use functions.
- Easy graphics.

Why is programming important?

- It gives you the power to do whatever you want with your data / experiment / simulations, without being limited by off-the-shelf software
 - It significantly enhances your capabilities as researchers.
- Some of the commonly used software in research are Matlab-based (like SPM for fMRI/MEG data analysis).

This talk

- Matlab basics
 - Variables
 - Matlab programming environment
 - Editor, scripts, functions

- Practice practice practice!

Variables

- A variable: a **place** in memory with a **name** that contains a **value**.
- Variables types – 2 basic types in Matlab (roughly speaking):
 - **Numeric**: single element (scalar), array, multi-dimensional array.
 - **Text**: character, string (array of characters).




A few words about syntax

- **Syntax** is “the **set of rules** that define the **combinations of symbols** that are considered to be correctly structured programs in a programming language”.
- In other words, it is the **vocabulary and grammar** with which we write our code, such that it will be unambiguously understandable by the programming language.
 - When defining a variable, refer to it later *exactly* in the same name. Tip: use **copy & paste**.
 - **Typos** are unacceptable.
 - Matlab is **case-sensitive**.

Defining numeric variables

- `x = 1 ;` (scalar, integer)
- `numSubjects = 8;` (meaningful name)
- `myScalar = 1.1;` (scalar, rational (decimal) number)
- `myVec = [1 2 3];` (one-dimensional array)
- `myVec = [1.2 2 3];` (one-dimensional array with mixed integers and rational (decimal) numbers)

Semicolon (;) at the end of a command prevents echo in the command line



Arrays and indexing

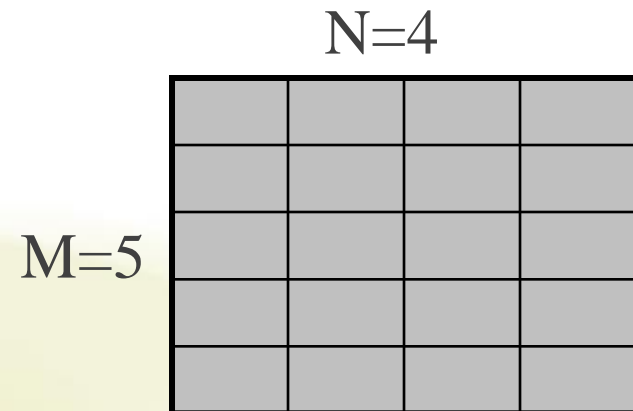
- **Array** – a set of ordered elements.
- **Indexing:** Every element in the array has a place called **index**.
 - The i-th element is the element in the i-th place.
- Defining arrays:
 - `myVec = [3 1 7 9 4];` → the index of 7 is 3
- **Retrieval** – Getting an element from a specific index in the array.
 - `arrayName(index)`
 - `myVec(3)` → 7
- **Assignment** – an element can be replaced:
 - `arrayName(index) = newValue`
 - `myVec(3) = 5` → `myVec = [3 1 5 9 4]`

<code>myVec(1)</code>	<code>myVec(2)</code>	<code>myVec(3)</code>	<code>myVec(4)</code>	<code>myVec(5)</code>
3	1	7	9	4

Matrices

- Matrix – 2D array (table).
 - Elements are ordered in 2 dimensions: rows and columns.
- M x N matrix – M rows, N columns.
- Example:
 - `myFirstMat = [1 2 3; 4 5 6];`

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$



Matrices - indexing

- Indexing: the a_{ij} element is the element in the **i-th** row and the **j-th** column.

- Example:

$$\begin{pmatrix} 5 & 8 & 12 & 4 \\ 7 & 1 & 9 & 3 \\ 11 & 5 & 2 & 13 \\ 3 & 6 & 10 & 8 \end{pmatrix}$$

a_{23}

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Arrays

- **Vector – one-dimensional** array.
 - Row: $1 \times N$ array.
 - Column: $N \times 1$ array
- **Matrix – two-dimensional** array
 - Table with rows and columns: $M \times N$
- **Three-dimensional** array/matrix – a kind of a cube of elements.
 - Dimensions $M \times N \times K$.
- **Multi-dimensional** arrays/matrices...

- All these arrays are simply **the same** data-type in Matlab, with just **different dimensions**, or size.

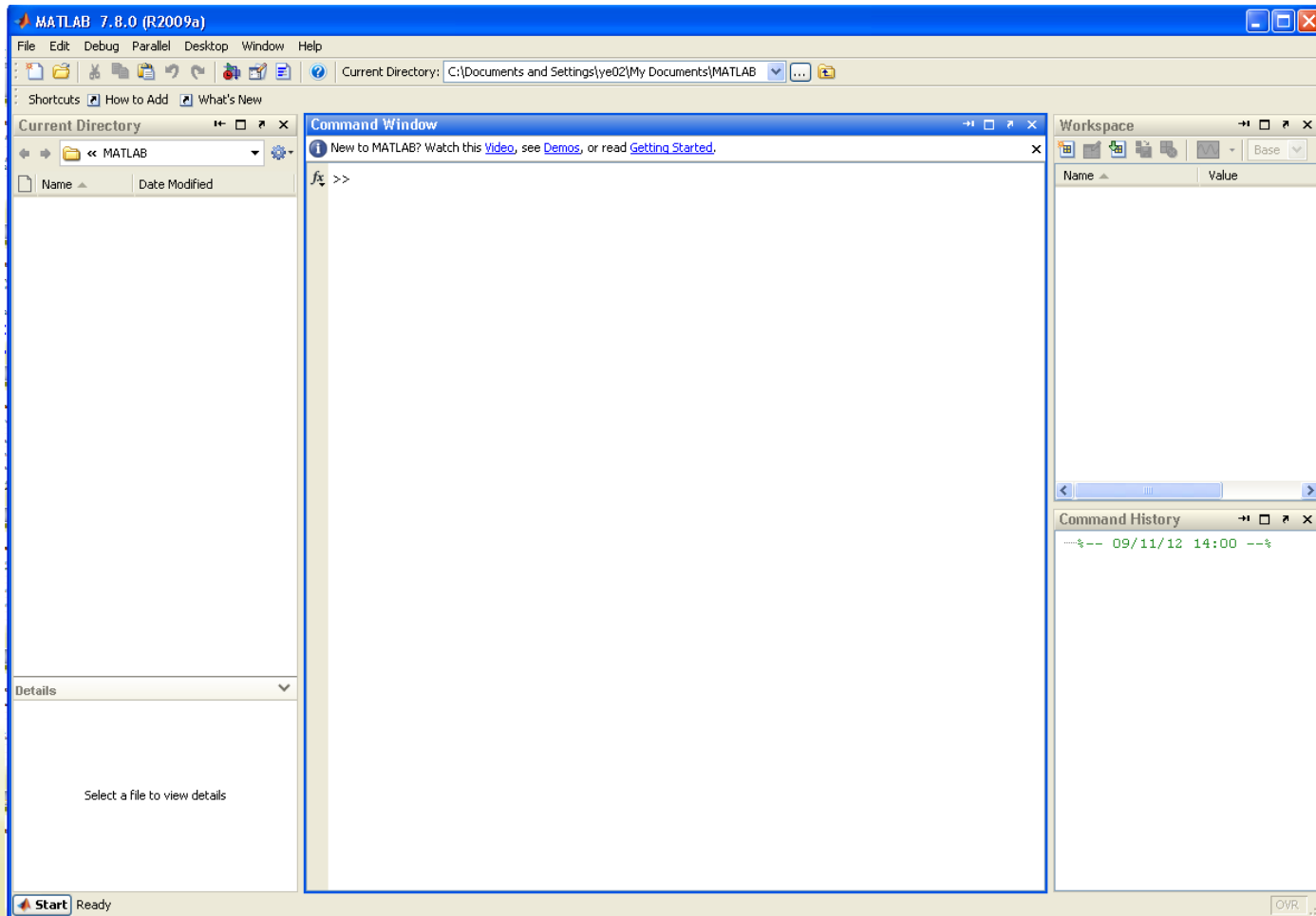
A few more notes about arrays

- We can assign values to a variable directly to its place in the array.
 - `myVar(2,3) = 5;`
 - `myVar([1 2],3) = [5 6];`
- We can assign a value of one variable to another variable
 - `x(2) = y;`
- We can delete an element from an array
 - `myVec = [1 2 3 4];`
 - `myVec(2) = [];`
 - `myVec([2 3]) = [];`

Text variables

- Text variables are comprised of **characters** and marked with “”.
 - myChar = 'h';
 - myChar = '5'; (this is not the number 5 but the character 5)
- A text variable can contain more than one character → **string** (an array of characters).
 - firstString = 'hello';
 - secondString = 'world';
 - longerOne = 'hello world';
 - longerOne2 = [firstString secondString]; (what's wrong with that?)

Matlab programming environment



'Current Directory'

- The **current directory** is the directory, or path, to which Matlab currently refers when reading/writing files, unless a different path is specified for a file.
- When opening Matlab, it is recommended to change the **current directory** to the one that you are working with.
 - It makes it easier to manage/find/save files.

Basic functions

- Matlab has a HUGE number of ready-to-use functions/commands. These are very useful and one of the major advantages of Matlab.
 - Examples: length, size, pwd, clc, clear, disp, sum, mean, std, zeros, rand, randn, save, load, and many more...

Code files

- Matlab code files have a **‘.m’** extension.
- They include the lines of code.
- Use the Matlab **editor** to edit and run code files
 - Scripts
 - Functions
 - More on that in the next talk by Jason.

Help!

- help *name_of_function*
- lookfor *keyword*
- helpdesk
- Internet

Exercise 1

- In the command window, do the following:
 - Create a 1x5 array with numeric values as you like.
 - Find the variable in the workspace and double-click it to see its content.
 - Change the value of the 3rd element in the array. Make sure you can see this change in the workspace.
 - Delete the 4th element in the array.
 - Use 'size' function to check for the size of the array.
 - Use 'length' function to check for the length of the array.
 - Clear all the variables and command window using 'clear' and 'clc'.

Exercise 2

- Create a Matlab code file and save it in your current directory. In this file, do the following:
 - Create a 3x4 matrix with values as you like.
 - Change the value of the element in the 2nd row and 3rd column.
 - Change all the values in the 2nd column at once by assigning a new vector.
 - Swap columns 1 and 3.
 - Delete the 4th column.
 - Use 'size' function to check for the size of the matrix.

Exercise 3

- In the command window, do the following:
 - Create a text variable that contains one word.
 - Create another text variable that contains one or more words.
 - Concatenate the two strings to create a third variable.
 - Display one of the strings in the command window using 'disp' function.